

Formularios Web con HTML y PHP

Índice de contenido

| | |
|---|----|
| 1 Intro..... | 1 |
| 2 Elemento <form> de un formulario HTML..... | 2 |
| 3 Campos de formularios HTML (controles)..... | 2 |
| 4 Campos de texto..... | 2 |
| 5 Botones de envío..... | 2 |
| 5.1 Ejemplo1. Mi color preferido..... | 3 |
| 5.2 Ejercicio1. Mis favoritos..... | 6 |
| 6 Casillas de verificación..... | 6 |
| 6.1 Ejemplo2. Casillas de verificación. ¿has ido a...? | 7 |
| 6.2 Ejercicio2. Ingredientes extras de una Pizza..... | 8 |
| 7 Botones de opción..... | 9 |
| 7.1 Ejemplo3. Botones de Opción. Test aritmética..... | 10 |
| 7.2 Ejemplo3 bis. Test aritmética Corregido. IF...ELSE..... | 12 |
| 7.3 Ejercicio3. Botones de opción. Examen C. Naturales..... | 13 |
| 8 Trabajo Final: Menú del Día..... | 14 |
| 9 OUTRO..... | 14 |

1 Intro

Existen dos formas básicas de interactuar con una página web, hacer clic sobre un enlace o enviar datos en un formulario.

Los formularios se usan en la navegación habitual por páginas web (i.e. Cuando entramos en nuestro correo web, iniciamos sesión en *youtube* para subir un vídeo, buscamos una página con *google*, o nos identificamos en *facebook* para acceder a nuestra red social,... siempre estamos enviando datos mediante los campos de un formulario.



Los formularios de una página web se diseñan como el resto de la página, empleando HTML (Lenguaje de Etiquetas de HiperTexto).

2 Elemento <form> de un formulario HTML

Para crear un formulario basta insertar etiquetas **<form>** de apertura y cierre en tu página WEB. Todos los controles (campos de texto, casillas de verificación, botones de opción o botón de enviar) que añadas entre las etiquetas **<form>** y **</form>** formarán parte del formulario que se envía al servidor WEB.

El elemento **<form>** cuenta con muchos atributos pero basta con utilizar 2 de ellos: **action** y **method**.

Atributo **action**: indica al servidor a qué página debe desplazarse cuando el usuario pulsa el botón de envío del formulario. Es decir, indica qué página recibirá la información del formulario. Por supuesto, esta página debe existir en el servidor Web.

Atributo **method**: controla la forma en que se envía la información al servidor y existen dos métodos, GET y POST. Con GET se envía la información introducida por el usuario adjuntándola a la URL después del signo “?”. Con POST se envían los valores introducidos en el formulario en el cuerpo de la solicitud HTTP.

El código siguiente crea un formulario WEB ... que aún no contiene ningún campo. Cuando añadamos los campos normales en un formulario, el archivo “**respuesta1.php**” recibirá mediante el método POST los valores introducidos en el formulario.

```
<form action="respuesta1.php"method="post">
</form>
```

3 Campos de formularios HTML (controles)

Varios son los controles que puedes utilizar para pedir información en un formulario. Para todos los ejemplos necesitaremos dos páginas Web. La primera es un formulario donde el usuario introduce la información solicitada en campos y la segunda recupera los valores del formulario para tratarlos posteriormente (i.e. mostrarlos en Tablas). Comenzaremos con los controles de formulario HTML más comunes:

4 Campos de texto

También llamados cuadros de texto, son probablemente los controles más conocidos en cualquier formulario. Se crean por medio del elemento **<input>** y estableciendo el atributo **type** en **text**.

```
<input type="text" name="direccion_de_trabajo">
```

En este ejemplo solicitamos la “dirección de trabajo” al usuario. Estos campos de texto aceptan frases de texto completas, lo que les hace muy interesantes para preguntas abiertas (i.e. Dirección donde trabajas).

5 Botones de envío

Los botones de envío se usan siempre en todos los formularios. Son necesarios para enviar la información del formulario a la página web receptora. Se crean por medio del elemento **<input>** y estableciendo el atributo **type** en **submit**. El texto que aparece sobre el botón se indica mediante el atributo **value**.

```
<input type="submit" value="Envíalo YA">
```

Este botón se mostraría como se ve a continuación:



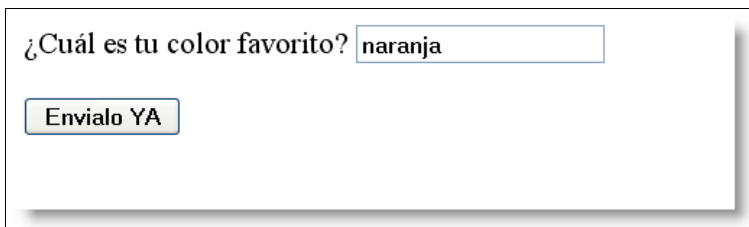
5.1 Ejemplo1. Mi color preferido

Vamos a crear un formulario que te pregunte por tu color preferido y luego lo muestre en pantalla. Recuerda, necesitamos 2 páginas web, una para el formulario y otra para mostrar el valor introducido. Para ello abre tu editor de código favorito (Kompozer, DevPHP, Bluefish,...) e introduce el siguiente código:

```
<html>
<head>
<title>Ejemplo 1. Formulario: Mi Color Favorito</title>
</head>
<body>
<form method="GET" action="respuesta1.php">
¿Cuál es tu color favorito?
<br>
<input type="text" name="color">
<br>
<input value="Envíalo YA" type="submit"><br>
</form>
</body>
</html>
```

Guárdalo en tu carpeta raíz como **ejemplo1.html** y cierra el archivo.

Cuando abrimos el archivo con un navegador web vemos el formulario:



Aún no funciona el botón de “enviar” ya que no hemos creado la segunda página web, la que recibe los valores introducidos en el formulario.

Como se ha indicado en el formulario `<form action="respuesta1.php">` sabemos que la segunda página se llamará “**respuesta1.php**”, evidentemente siempre podemos cambiar el nombre del archivo en el atributo **action**.

¿Cómo podemos recuperar la información del color introducido en el formulario? Sólo con HTML NO BASTA, necesitamos escribir guiones con PHP, un lenguaje de programación que trabaja del lado del servidor de páginas web.

En las próximas líneas vamos a abrir una pequeña ventana al lenguaje PHP, si tienes dudas de mi explicación, hay un tutorial de PHP4 divertido y muy, muy claro que he encontrado en <http://www.rinconastur.net/> dentro de la sección de informática.

Seguimos con PHP, estos guiones o líneas de código se introducirán dentro del código HTML de la página web, y para distinguirlas irán siempre delimitadas por los caracteres '`<?php`' y '`?>`'.

El código de nuestra página es el siguiente, introdúcelo en tu editor web y guárdalo como **respuesta1.php**

```
<body>
Tu color favorito es: <br>
<?php echo ($_GET['color']);?>
</body>
</html>
```

¿Que hace nuestro guión PHP?

`<?php` indica que iniciamos un guión escrito en php.

echo es un comando que muestra en la página web las palabras (cadenas de texto) y/o variables. Las cadenas de texto deben ir entrecomillas, las variables no lo necesitan. Para que sea más claro el comando **echo** usamos paréntesis () para delimitar el contenido que vamos a mostrar..

En nuestro ejemplo el comando **echo** mostrará en la página web el contenido de la variable `$_GET['color']`,...pero ¿qué es esto? No es difícil, ahora te lo explico...

`$_GET['color']` es una variable automáticamente creada por PHP y que almacena el valor del 'color' introducido en el campo de texto con ese nombre. ¿un poco rara? bueno, ya te acostumbrarás... tiene varios símbolos raros que también te explico:

- `$` se emplea siempre en PHP para indicar que lo que viene a continuación es una variable.
- `_GET[]` porque el método de envío del formulario elegido es GET, y los corchetes que contendrán el nombre del campo de texto.
- `'color'` porque es el nombre que le dimos al campo de texto en el formulario, y las comillas sencillas `'` se usan siempre que trabajamos con cadenas alfanuméricas, es decir, textos.
- `“;”` El punto y coma siempre se usa al final de la instrucción PHP, es imprescindible ponerla.

**Atento: PHP es superquisquilloso con las mayúsculas y minúsculas, para él 'A' y 'a' son distintas: si el nombre de nuestro campo de texto es 'color', no podemos invocarlo en PHP como 'Color', ni así 'coLor',....¡ATENCIÓN!.*

Ahora ya tenemos las dos páginas web, el formulario donde nos hacen la pregunta del color preferido con un campo de texto para responderla y la página **respuesta1.php** que mostrará el color elegido. Pero si abres con un navegador el formulario y lo envías pulsando el botón, **NO APARECE TU RESPUESTA** “naranja” en página **respuesta1.php**...¿por que?

Tu color favorito es:

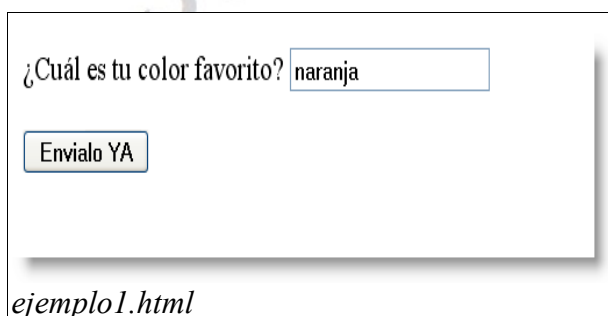
IMPORTANTE: Bien, si te fijas el archivo **respuesta1.php** tiene extensión PHP estos archivos sólo se pueden interpretar en el servidor de páginas web. Bien pero, ¿Quién tiene un servidor de páginas web en su PC?...

Hoy en día esto es fácil y... gratuito. El servidor web más usado en el mundo es APACHE, tanto en Windows como en Linux. En Windows existen varias *suites* gratuitas que instalan APACHE con un sólo click en tu PC, junto con servidores de bases de datos MySQL, servidor de Correo Mercury y servidores FTP Filezilla.

Una de las *suites* más conocidas es XAMPP, si lo haces con ella todas las páginas web deben guardarse en la carpeta "**c:\XAMPP\htdocs**", para que nos las sirva el servidor APACHE. En Linux, y concretamente en la distribución Ubuntu (MAX es Ubuntu), tenemos ya instalado un servidor APACHE, y la carpeta donde debemos guardar las páginas web es "**/var/www/**".

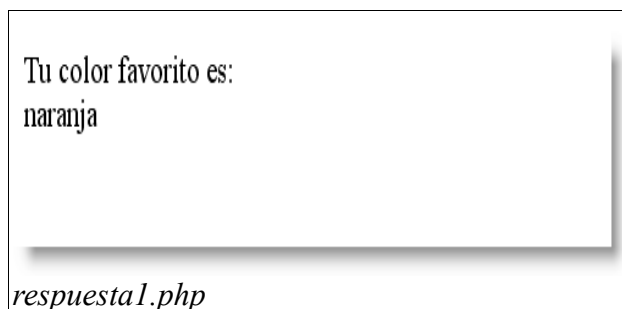
MÁS IMPORTANTE: nosotros siempre vamos a copiar todas nuestras páginas web, sean HTML o PHP al ordenador del profesor que cuenta con un servidor APACHE, a una carpeta compartida que encontrareis en **profesor:/miweb/PC1/** . APACHE servirá las páginas web que haya en esta dirección ya que esta dirección está enlazada con **/var/www/** .

AÚN MÁS IMPORTANTE: Y si quiero ver de una vez mi formulario en acción...¿Que HAGGOOO? Bien, abre el navegador web (i.e. Mozilla Firefox) en tu PC y en el campo de texto de las direcciones URL introduce la dirección donde está el formulario: i.e. "**dirección_IP_local_PC_profesor/miweb/PC1/**", donde dirección IP local del profesor será del tipo 192.168.1.116 u otra similar, y verás las páginas web que el servidor APACHE puede servirte desde esa dirección. Haz click en el **ejemplo1.html**, verás el formulario que has diseñado. Rellena la respuesta y envíalo. Verás tu color favorito escrito en la página **respuesta1.php** .



¿Cuál es tu color favorito?

ejemplo1.html



Tu color favorito es:
naranja

respuesta1.php

5.2 Ejercicio1. Mis favoritos

Te propongo un ejercicio sin mostrarte la solución, así comprobarás si has aprendido bien cómo crear tu primer formulario. Ánimo...

Tarea: Crea un formulario con los campos de texto que ves en la captura y luego crea una sencilla página para mostrar los valores introducidos.

ejercicio1.html

ejecicio1.php

Si has hecho bien el `ejercicio1` enséñaselo al profesor y continua con el tutorial.

6 Casillas de verificación

Es una sola casilla que se puede marcar o nó, según lo que seleccionemos. Para crearla el elemento `<input>` debe tener como atributo `type="checkbox"`, además si queremos que aparezca de entrada marcada la casilla lo indicaremos con `checked`. Como sólo podemos marcar o desmarcar la casilla es ideal para respuestas del tipo SI/NO. Para que pase alguna información más útil lo indicaremos en el atributo `value` (i.e. `value="Museo del Prado"`).

6.1 Ejemplo2. Casillas de verificación. ¿has ido a...?

Vamos a crear un formulario que pregunte si has estado en un par de museos...

ejemplo2.html

Para ello el código HTML es:

```
<body>
<form method="get" action="ejemplo2.php">
Nombre: <input name="nombre"> y apellidos: <input name="apellidos">
<br>
<br>
¿alguna vez has ido....
<br>
al museo del Prado? <input name="prado" value="Museo del Prado" type="checkbox">
<br>
al Oceanografic de Valencia? <input name="oceanografic"
value="Oceanográfico de Valencia" type="checkbox">
<br>
<input value="nViaDo" type="submit"><br>
</form>
</body>
```

Guarda el archivo como **ejemplo2.html** y a continuación vamos a crear la página donde se presentará la información introducida en el formulario.

El archivo **ejemplo2.php** recogerá las respuestas introducidas en las casillas y mostrará sus valores (**value="Museo del Prado"** y **value="Oceanográfico de Valencia"**). También mostrará el nombre y apellidos separados por un espacio entre ellos con esta línea:

```
<?php echo ($_GET['nombre']." ".$_GET['apellidos']);?>
```

Fíjate bien que después del nombre se añade entre comillas un espacio y luego se añade el apellido. Y fíjate que se usa el punto "." para concatenar cadenas en la instrucción **echo**.

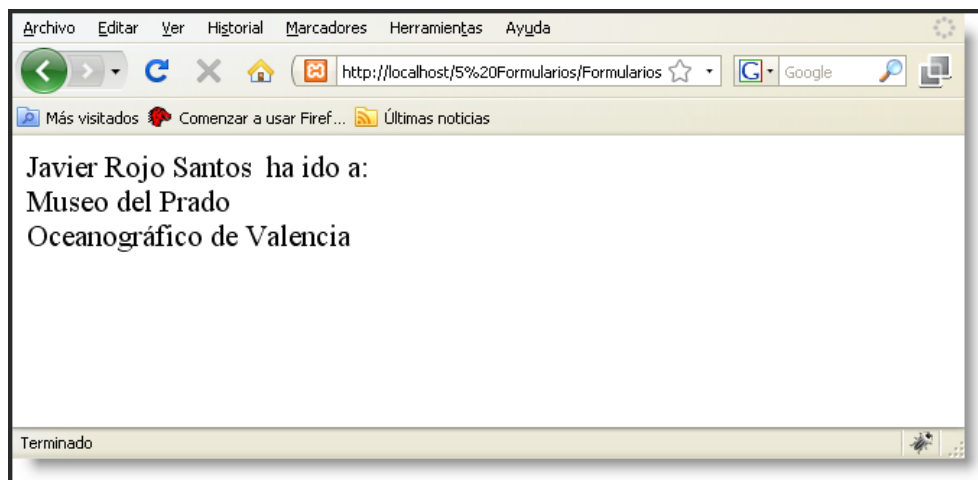
Esto significa que se mostrarán juntos en la página web como una sola línea los tres bloques

separados por “.”, `$_GET['nombre']` seguido de un espacio y luego `$_GET['apellidos']`

El código completo se muestra a continuación, introdúcelo en tu editor web y guárdalo como `ejemplo2.php`.

```
<body>
<?php echo ($_GET['nombre']." ".$_GET['apellidos']);?>
ha ido a: <br>
<?php echo ($_GET['prado']);?>
<br>
<?php echo ($_GET['oceanografic']);?>
<br>
</body>
```

Se debería ver en un navegador como en la captura siguiente:



ejemplo2.php

Si todo ha ido bien estás preparado para hacer el ejercicio2...

6.2 Ejercicio2. Ingredientes extras de una Pizza

Tarea: Crea un formulario que pregunte si el cliente de una pizzería quiere o no ciertos ingredientes extras. Fíjate en la siguiente captura:

PIZZERÍA "Piemonte"

| | |
|----------|----|
| Mesa nº: | 24 |
|----------|----|

Ha pedido una pizza Especial de la Casa .

Desea alguno de los ingredientes siguientes:

Pimiento: ☒

Queso Gorgonzola: ☐

Rúcola: ☒

Terminado

ejercicio2.html

La mesa nº 24 ha pedido los siguientes extras para su pizza:

pimiento

rúcola

Terminado

ejercicio2.php

Cuando lo termines y funcione, muéstraselo a tu profesor...

7 Botones de opción

Consiste en un círculo que se puede seleccionar o no. Normalmente se usan varios botones juntos y SÓLO se puede seleccionar uno de ellos. Es ideal para las respuestas tipo TEST con una sola respuesta posible de entre varias opciones. Los botones de opción se crean con el elemento `<input>` y su atributo `type="radio"`. Es importante que todos los botones "conectados" en el mismo grupo tengan el mismo nombre (atributo `name`). Además, si queremos que uno de los botones aparezca de entrada como marcado se usa el atributo `checked`. La información que se pasa a la segunda página se puede controlar con el atributo `value="valor que quiera enviar al seleccionar"`.

este botón”, observa el siguiente ejemplo.

7.1 Ejemplo3. Botones de Opción. Test aritmética

Vamos a crear un formulario con una pregunta tipo test sobre aritmética. Tendrá 4 respuestas posibles y sólo se podrá elegir una de ellas. Se mostrarán siempre por defecto la primera opción como marcada (necesitaremos usar el atributo **checked**). En la siguiente captura se muestra el formulario:

Examen tipo test de aritmética

Nombre: Apellidos:

1. ¿ Cuanto es $3+(3 \times 6)$?

a) 17 ☒

b) 21 ☐

c) 16 ☐

d) Ninguna de las anteriores respuestas ☐

ejemplo3.html

Introduce este código HTML en tu página Web y guárdalo en tu carpeta raíz como “ejemplo3.html”:

```
<body>
<h2>Examen tipo test de aritmetica</h2>
<form method="get" action="ejemplo3.php">
Nombre: <input name="nombre"> Apellidos: <input name="apellidos">
<br>
<br>
1. Cuanto es 3+(3x6) ?
<br>
a) 17 <input checked="checked" name="aritmetica" value="17" type="radio">
<br>
b) 21 <input name="aritmetica" value="21" type="radio">
<br>
```

```

c) 16 <input name="aritmetica" value="16" type="radio">
<br>
d) Ninguna de las anteriores respuestas<input name="aritmetica" value="ninguna de
las respuestas mostradas" type="radio">
<br>
<br>
<input value="nViaDo" type="submit"><br>
</form>
</body>

```

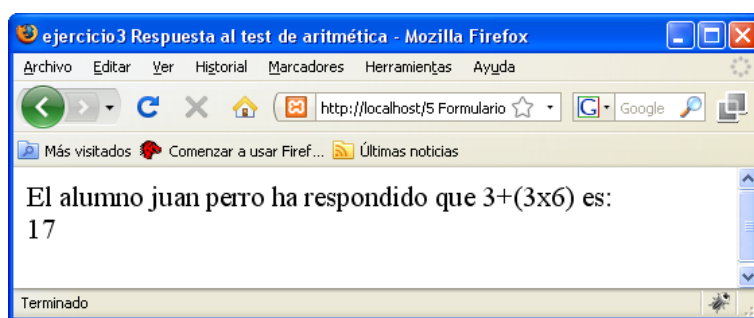
Explicación del código.

Como ya tienes cierto conocimiento en el manejo de formularios iré a lo más importante:

1. Al crear el formulario se muestra el atributo **action** con el nombre del archivo (**ejemplo3.php**) donde se enviará la información del formulario.
2. Fíjate que todos los botones de opción tienen el mismo atributo nombre (**name="aritmetica"**) de esta forma el formulario reconoce que los botones pertenecen al mismo grupo y que sólo uno de ellos puede ser seleccionado a la vez.
3. En cada botón se define el atributo **value** igual al valor que se mostrará en pantalla ("17", "21", etc) al enviar la información del formulario.

IMPORTANTE: Si hiciéramos un examen tipo test con varias preguntas sólo habría que nombrar cada grupo de botones como **name="respuesta1"**, **name="respuesta2"**, y así sucesivamente.

Ahora vamos con la página llamada "**ejemplo3.php**" que recoge la información introducida en el formulario. La captura siguiente te la muestra.



ejemplo3.php

El código siguiente consigue recuperar la información del formulario y mostrarla en pantalla. Introdúcelo en tu editor web y guárdalo en tu carpeta raíz como "**ejemplo3.php**".

```

<body>
El alumno <?php echo $_GET['nombre']. " ".$_GET['apellidos'];?> ha respondido que
3+(3x6) es:

```

```
<br>
<?php echo $_GET['aritmetica'];?>
<br>
<br>
</body>
```

No hay mucho que explicar, el primer guión PHP muestra en pantalla mediante el comando **echo** el nombre y apellidos, como ya hemos visto en otros ejemplos. Y la línea **<?php echo \$_GET['aritmetica'];?>** muestra el valor del botón (atributo **value="17"**) que se ha seleccionado.

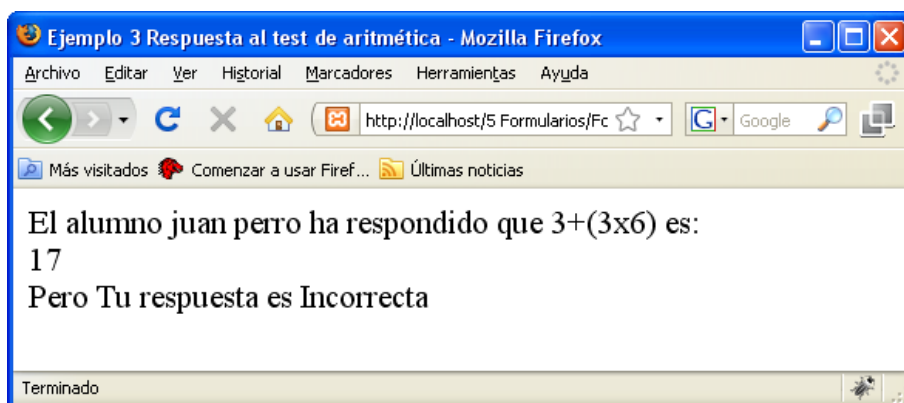
Después de hacer este ejemplo sobre botones de opción para hacer preguntas tipo test, uno se hace siempre la misma pregunta. **¿Podríamos ver si la respuesta seleccionada es correcta?** Por supuesto que sí, pero para ello necesitamos manejar otros comandos de PHP, las sentencias **IF...ELSE**. Son sencillas de emplear, pero su manejo se escapa a las pretensiones de este tutorial. Aún así, si quieres, prueba a añadir este código en tu página **ejemplo3.php**

7.2 Ejemplo3 bis. Test aritmética Corregido. IF...ELSE

Este ejemplo es **VOLUNTARIO, VALE PARA SUBIR NOTA...** si se comprende..
Añade el código PHP a tu página **ejemplo3.php** y guárdalo como **ejemplo3bis.php**.

```
<body>...
<?php echo $_GET['aritmetica'];
if ($_GET['aritmetica']==21) {echo ("Tu respuesta es correcta");}
else {echo ("<br> Pero Tu respuesta es Incorrecta");}
?>
</body>
```

La sentencia **IF...ELSE** (SI...SI NO) funciona así: **SI** (se cumple esta condición) {hago esto} **SI NO** {hago lo otro}. Si la variable **\$_GET['aritmetica']** es igual a 21 muestra en pantalla “Tu respuesta es correcta” si no muestra “Pero Tu respuesta es Incorrecta”.



ejemplo3bis.php

Si te parece muy difícil **NO** te PREOCUPES, no es nuestro objetivo aprender estas **estructuras de toma de decisiones**.

7.3 Ejercicio3. Botones de opción. Examen C. Naturales

Tarea: Crea un examen de Ciencias Naturales como se muestra en el siguiente formulario y guárdalo como **ejercicio3.html**.

Examen tipo test de C. Naturales

Nombre: Apellidos:

1. ¿ Cuantos planetas hay en nuestro sistema solar?

- a) 7 planetas ☒
- b) 8 planetas ☐
- c) 9 planetas ☐
- d) Ninguna de las anteriores ☐

2. Los quelonios son:

- a) Ranas ☐
- b) Culebras ☒
- c) Tortugas ☐
- d) Ninguna de las anteriores ☐

Terminado

ejercicio3.html

EL archivo que recoge las respuestas se llamará **ejercicio3.php**, a continuación te muestro una captura:

Ejercicio 3 Respuesta al test de Ciencias Naturales

El alumno juan perro ha respondido que:

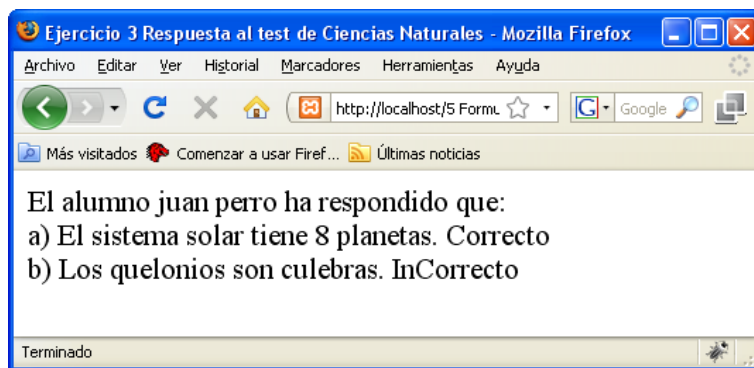
- a) El sistema solar tiene 7 planetas.
- b) Los quelonios son culebras.

Terminado

ejercicio3.php

Sólo si eres capaz, intenta que este archivo **ejercicio3.php** muestre si las respuestas son

correctas o nó, como se muestra en la captura siguiente. Guárdalo como **ejercicio3bis.php**



ejercicio3bis.php

8 Trabajo Final: Menú del Día

Crea un formulario que muestre el menú del día en un restaurante, tendrás que pedir los siguientes datos:

1. Número de la mesa
2. Nombre y apellidos del camarero que atiende la mesa
3. Tres primeros platos a elegir uno de ellos
4. Tres segundos platos a elegir uno de ellos
5. Tres bebidas a elegir una de ellas.
6. Extras como pan, mantequilla, puro pudiendo escoger los que quieras.

Guarda el archivo como **trabajo_final.html** en tu carpeta raíz

Crea otro archivo para mostrar el menú escogido y guárdalo como **“trabajo_final.php”** en la misma carpeta raíz.

9 OUTRO

Evidentemente, cualquiera que haya visto un formulario sabe que nos hemos saltado algunos controles muy usados, botones de RESET, listas desplegables, campos para contraseñas,.... pero NO es nuestro empeño verlos todos. Hay cientos de ejemplos en páginas web para aprender, ahora la pelota está en tu TEJADO.

También es evidente que con la información que se introduce en los formularios se puede hacer muchas más cosas que simplemente presentarlas en pantalla. Te he mostrado por una rendija pequeña el uso de PHP como corrector de exámenes, pero si piensas un poco, reconocerás que sólo cuando introduces tus datos correctos puedes comprobar tu correo, esto es una aplicación más PHP, los *login* o autenticaciones de usuario y así muchas, muchas mas. Como he dicho antes, si te gusta verás que es mucho más fácil de lo que te ha parecido hasta ahora, y es que los comienzos siempre son lo más difícil.